Sequential Randomized Algorithms for Convex Optimization in the Presence of Uncertainty

Mohammadreza Chamanbaz, Fabrizio Dabbene, Roberto Tempo,
Venkatakrishnan Venkataramanan, Qing-Guo Wang

Abstract

Motivated by the complexity of solving convex scenario problems in one-shot, in this paper we provide a direct connection between this approach and sequential randomized methods. A rigorous analysis of the theoretical properties of two new algorithms, for full constraint satisfaction and partial constraint satisfaction, is provided. These algorithms allow to enlarge the applicability of scenario-based methods to real-world applications involving a large number of design variables. Extensive numerical simulations for a non-trivial application regarding hard-disk drive servo design testify the goodness of the proposed solution.

I. Introduction

In recent years, research on randomized and probabilistic methods for control of uncertain systems has successfully evolved along various directions, see e.g. [19] for an overview of the state of the art on this topic. In particular, different approaches and techniques have been developed and tested in several applications, see e.g. [9]. For convex control design, two main classes of algorithms, sequential and non-sequential, have been proposed in the literature, and their theoretical properties have been rigorously studied.

Regarding non-sequential methods, the approach that has emerged is the so-called scenario approach, which has been introduced in [6], [7]. Taking random samples of the uncertainty

M. Chamanbaz and V. Venkatakrishnan are with Data Storage Institute, Singapore (emails: {Mohammad_C, Venka_V}@dsi.a-star.edu.sg).

F. Dabbene and R. Tempo are with CNR-IEIIT Torino, Italy (emails: {roberto.tempo, fabrizio.dabbene}@polito.it).

M. Chamanbaz and Q-G. Wang are with Department of Electrical and Computer Engineering, National University of Singapore (emails: {Chamanbaz,Elewqg}@nus.edu.sg).

$q \in \mathbb{Q}$, the main idea of this particular line of research is to reformulate a semi-infinite convex optimization problem as a sampled optimization problem subject to a finite number of random constraints. Then, a key problem is to determine the sample complexity, i.e. the number of random constraints that should be generated, so that the so-called probability of violation is smaller than a given accuracy $\epsilon \in (0, 1)$, and this event holds with a suitably large confidence $1 - \delta \in (0, 1)$. A very nice feature of the scenario approach is that the sample complexity is determined a priori, that is before the sampled optimization problem is solved, and it depends only on the number of design parameters, accuracy and confidence. On the other hand, if accuracy and confidence are very small, and the number of design parameters is large, then the sample complexity may be huge, and the sampled convex optimization problem cannot be easily solved in practice.

A parallel line of research, mainly focused on deriving sequential methods for feasibility, has been developed for various specific control problems, which include linear quadratic regulators, linear matrix inequalities and switched systems as particular cases of a general framework, based on various update rules and probabilistic oracles, presented in [9], [19]. The main idea of these sequential methods is to introduce the concept of validation samples. That is, at step k of the sequential algorithm, a “temporary solution” is constructed and, using a suitably generated validation sample set, it is verified whether or not the probability of violation corresponding to the temporary solution is smaller than a given accuracy ϵ, and this event holds with confidence $1 - \delta$. To study the properties of these algorithms, the sample complexity of the validation set should be derived, but in this case, unlike the scenario approach, the sample complexity is a random variable which cannot be derived a priori. Due to their sequential nature, these algorithms might have wider applications than the scenario approach, in particular in real-world problems where fast computations are needed because of very stringent time requirements due to on-line implementations. However, at present, most sequential algorithms studied in the literature are limited to probabilistic feasibility problems. One of the exceptions is the method based on stochastic bisection proposed in [21]. A general framework for nonconvex problems is introduced in [1], where the class of sequential probabilistic validation (SPV) algorithms is studied.

In this paper, which is an expanded version of [12], we study two new sequential algorithms for optimization with full constraint satisfaction and partial constraint satisfaction, respectively, and we provide a rigorous analysis of their theoretical properties regarding the probability of violation. These algorithms fall into the class of SPV algorithms, but exploit specific convexity and finite convergence properties of scenario methods, thus showing computational improvements upon those presented in [1], see Section III-A. In particular, the sample complexity of both algorithms is derived and it enters directly into the validation step. The sample complexity increases very mildly with probabilistic accuracy, confidence and number of design parameters, and depends on a termination parameter which is chosen by the user. In the worst case, an optimization problem having the same size of the scenario approach should be solved.

In the second part of the paper, using a non-trivial example regarding the position control of read/write head in a commercial hard disk drive, we provide extensive numerical simulations which compare upfront the sample complexity of the scenario approach with the number of iterations required in the two sequential algorithms previously introduced. We remark again that the sample complexity of the scenario approach is computed a priori, while for sequential algorithms, the numerical results regarding the size of the validation sample set are random. For this reason, mean values, standard deviation and other related parameters are experimentally computed for both proposed algorithms by means of extensive Monte Carlo simulations.

II. Problem Formulation and Preliminaries

An uncertain convex problem has the form

$$\begin{aligned} &\underset{\theta\in\Theta}{\text{minimize}} \quad c^T\theta \\ &\text{subject to } f(\theta,q)\leq 0 \text{ for all } q\in\mathbb{Q} \end{aligned} \qquad (1)$$

where $\theta\in\Theta\subset\mathbb{R}^{n_\theta}$ is the vector of optimization variables and $q\in\mathbb{Q}\subset\mathbb{R}^\ell$ denotes the vector of uncertain parameters bounded in the set $\mathbb{Q}$, $f(\theta,q):\Theta\times\mathbb{Q}\to\mathbb{R}$ is convex in θ for any fixed value of $q\in\mathbb{Q}$ and Θ is a convex and closed set. We note that most uncertain convex problems can be reformulated as (1). In particular, multiple scalar-valued constraints $f_i(\theta,q)\leq 0,\ i=1,\ldots,m$ can always be recast into the form (1) by defining $f(\theta,q)=\max_{i=1,\ldots,m} f_i(\theta,q)$.

In this paper, we study a probabilistic framework in which the uncertainty vector q is assumed to be a random variable and the constraint in (1) is allowed to be violated for some $q\in\mathbb{Q}$, provided that the rate of violation is sufficiently small. This concept is formally expressed using the notion of "probability of violation".

Definition 1 (Probability of Violation): The probability of violation of θ for the function f :

$\Theta \times \mathbb{Q} \to \mathbb{R}$ is defined as

$$V(\theta) \doteq \Pr\left\{q \in \mathbb{Q} \;:\; f(\theta, q) > 0\right\}. \tag{2}$$

The exact computation of $V(\theta)$ is in general very difficult since it requires the computation of multiple integrals associated to the probability in (2). However, this probability can be estimated using randomization. To this end, assume a probability measure is given over the set $\mathbb{Q}$, we generate N independent identically distributed (i.i.d.) samples within the set $\mathbb{Q}$

$$\mathbf{q} = \{q^{(1)}, \ldots, q^{(N)}\} \in \mathbb{Q}^N$$

based on the given density function, where $\mathbb{Q}^N \doteq \mathbb{Q} \times \mathbb{Q} \times \cdots \times \mathbb{Q}$ (N times). Next, a Monte Carlo approach is employed to obtain the so called "empirical violation" which is introduced in the following definition.

Definition 2 (Empirical Violation): For given $\theta \in \Theta$ the empirical violation of $f(\theta, q)$ with respect to the multisample $\mathbf{q} = \{q^{(1)}, \ldots, q^{(N)}\}$ is defined as

$$\widehat{V}(\theta, \mathbf{q}) \doteq \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_f(\theta, q^{(i)}) \tag{3}$$

where $\mathbb{I}_f(\theta, q^{(i)})$ is an indicator function defined as $\mathbb{I}_f(\theta, q^{(i)}) \doteq \begin{cases} 0 & \text{if } f(\theta, q) \leq 0 \\ 1 & \text{otherwise} \end{cases}$.
It is clear that, based on the definition of $\mathbb{I}_f(\theta, q^{(i)})$, the empirical violation is a random variable bounded in the closed interval $[0, 1]$.

A. The Scenario Approach

In this subsection, we briefly recall the so-called scenario approach, also known as random convex programs, which was first introduced in [6], [7], see also [10] for additional results. In this approach, a set of independent identically distributed random samples of cardinality N is extracted from the uncertainty set and the following random convex program is formed

$$\begin{aligned} &\underset{\theta \in \Theta}{\text{minimize}} \quad c^T \theta \\ &\text{subject to } \; f(\theta, q^{(i)}) \leq 0, \; i = 1, \ldots, N. \end{aligned} \tag{4}$$

The function $f(\theta, q)$ is convex for fixed $q \in \mathbb{Q}$ and a further assumption is that the problem (4) attains a unique solution $\widehat{\theta}_N$. These assumptions are now formally stated.

Assumption 1 (Convexity): $\Theta \subset \mathbb{R}^{n_\theta}$ is a convex and closed set and $f(\theta, q)$ is convex in θ for any fixed value of $q \in \mathbb{Q}$.

Assumption 2 (Uniqueness): If the optimization problem (4) is feasible, it admits a unique solution.

We remark that the uniqueness assumption can be relaxed in most cases by introducing a tie-breaking rule (see Section 4.1 of [6]). The probabilistic property of the optimal solution obtained from (4) is stated in the next lemma; see Theorem 1 in [7].

Lemma 1: Let Assumptions 1 and 2 hold and let δ, $\varepsilon \in (0, 1)$ and N satisfy the following inequality

$$\sum_{i=0}^{n_\theta - 1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \leq \delta. \tag{5}$$

Then, either the optimization problem (4) is infeasible which means the original problem (1) is also infeasible or, if feasible, with probability no smaller than $1-\delta$, its optimal solution $\widehat{\theta}_N$ satisfies the inequality $V(\widehat{\theta}_N) \leq \varepsilon$.

There are a number of results in the literature for deriving bounds on the smallest sample complexity N which satisfies (5). The least conservative one, which is proved in [3], is stated here.

Lemma 2: Let Assumptions 1 and 2 hold. Then, for given $\varepsilon \in (0, 1)$ and $\delta \in (0, 1)$, Lemma 1 holds if

$$N \geq \inf_{a>1} \frac{1}{\varepsilon} \left(\frac{a}{a-1}\right) \left(\ln \frac{1}{\delta} + (n_\theta - 1) \ln a\right). \tag{6}$$

B. Scenario with Discarded Constraints

The idea of scenario with discarded constraints [5], [11] is to generate N i.i.d. samples and then purposely discard $r < N - n_\theta$ of them. In other words, we solve the following optimization problem

$$\begin{aligned} &\underset{\theta \in \Theta}{\text{minimize}} \quad c^T \theta \\ &\text{subject to} \;\; f(\theta, q^{(i)}) \leq 0, \, i = 1, \ldots, N-r. \end{aligned} \tag{7}$$

The r discarded samples are chosen so that the largest improvement in the optimal objective value is achieved. We remark that the optimal strategy to select r discarded samples is a mixed-integer optimization problem, which may be hard to solve numerically. The following lemma [5] defines the probabilistic properties of the optimal solution obtained from (7).

Lemma 3: Let Assumptions 1 and 2 hold and let $\delta,\, \varepsilon \in (0,1)$, N and $r < N - n_\theta$ satisfy the following inequality

$$\binom{r+n_\theta}{r} \sum_{i=0}^{r+n_\theta} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \leq \delta. \tag{8}$$

Then, either the optimization problem (7) is infeasible which means the original problem (1) is also infeasible or, if feasible, with probability no smaller than $1-\delta$, its optimal solution $\widehat{\theta}_N$ satisfies the inequality $V(\widehat{\theta}_N) \leq \varepsilon$.

An explicit sample bound N satisfying (8) is also reported in [5].

Lemma 4: Let Assumptions 1 and 2 hold. Then, for given $\varepsilon \in (0,1)$, $\delta \in (0,1)$ and r, Lemma 3 holds if

$$N \geq \frac{2}{\varepsilon} \ln \frac{1}{\delta} + \frac{4}{\varepsilon}(r + n_\theta). \tag{9}$$

The sample bounds (6) and (9) can be very large even for problems with moderate number of decision variables. Therefore, the computational complexity of the random convex problems (4) and (7) might be beyond the capability of the available computational tools. Motivated by this limitation, in the next section we propose two sequential randomized algorithms.

III. The Sequential Randomized Algorithms

The main philosophy behind the proposed sequential randomized algorithms lies on the fact that it is easy from the computational point of view to evaluate a given "candidate solution" for a large number of random samples extracted from $\mathbb{Q}$. On the other hand, it is clearly more expensive to solve the optimization problems (4) or (7) when the sample bound N is large. The sequential randomized algorithms, which are presented next, mitigate the conservativeness of the bounds (6) and (9) by generating a sequence of "design" sample sets $\{q_d^{(1)}, \ldots, q_d^{(N_k)}\}$ with increasing cardinality N_k which are used in (4) and (7) for solving the optimization problem. Parallel "validation" sample sets $\{q_v^{(1)}, \ldots, q_v^{(M_k)}\}$ of cardinality M_k are also generated by both algorithms in order to check whether the given candidate solution, obtained from solving (4) or (7), satisfies the desired violation probability.

The first algorithm is in line with those presented in [8] and [17], in the sense that it uses a similar strategy to validate the candidate solution. However, while these algorithms have been designed for feasibility problems, the proposed algorithms deal with optimization problems.

More generally, the two presented algorithms fall into the class of general SPV algorithms studied in [1].

A. Full Constraint Satisfaction

The first sequential randomized algorithm is presented in Algorithm 1, and its theoretical properties are stated in the following theorem.

Theorem 1: Suppose that Assumptions 1 and 2 hold. If at iteration k Algorithm 1 exits with a probabilistic solution $\widehat{\theta}_{N_k}$, then it holds that $V(\widehat{\theta}_{N_k}) \leq \varepsilon$ with probability no smaller than $1-\delta$

$$\Pr\left\{V(\widehat{\theta}_{N_k}) \leq \varepsilon\right\} \geq 1-\delta.$$

Proof: See Appendix A. ■

Remark 1 (Optimal Value of α): The sample bound (11) is similar to the one derived in [9, Theorem 2] originally proven in [15], and also used in [1]. However, since we are using a finite sum[1], thanks to the scenario bound (6), we can use a finite hyperharmonic series (also known as p-series) instead of the Riemann Zeta function. The Riemann Zeta function does not converge when α is smaller than one, while in the presented bound (11) α may be smaller than one, which improves the sample complexity in particular for large values of k_t. The optimal value of α which minimizes the sample bound (11) has been computed using numerical simulations for different values of the termination parameter k_t. The almost optimal value of α minimizing (11) for a wide range of k_t is $\alpha = 0.1$. The bound (11) (for $\alpha = 0.1)$ improves upon the bound (17) in [9], by 5% to 15% depending on the termination parameter k_t. It also improves upon the bound in [17], which uses finite sum but in a less effective way.

B. Partial Constraint Satisfaction

In the “design” and “validation” steps of Algorithm 1, *all* elements of the design and validation sample sets are required to satisfy the constraint in (1). However, it is sometimes impossible finding a solution satisfying the constraint in (1) for the entire set of uncertainty. In Algorithm 2, we consider the scenario design with discarded constraints where we allow a limited number of design and validation samples to violate the constraint in (1). We now state a theorem explaining the theoretical properties of Algorithm 2.

[1] See in particular the summation (16) in the proof of Theorem 1.

Algorithm 1 SEQUENTIAL RANDOMIZED ALGORITHM: FULL CONSTRAINT SATISFACTION

1) INITIALIZATION

   Set the iteration counter to zero $(k = 0)$. Choose the desired probabilistic levels ε, δ and the desired number of iterations $k_t > 1$.

2) UPDATE

   Set $k = k + 1$ and $N_k = \left\lceil N \frac{k}{k_t} \right\rceil$ where N is chosen based on (6) and $\lceil x \rceil$ denotes the smallest integer greater than or equal to x.

3) DESIGN

   - Draw N_k i.i.d. samples $\mathbf{q}_d = \{q_d^{(1)} \dots q_d^{(N_k)}\}$ from the uncertainty set $\mathbb{Q}$ based on the underlying distribution.
   - Solve the following random convex program

   $$\begin{aligned} \widehat{\theta}_{N_k} = & \arg \underset{\theta \in \Theta}{\text{minimize}} \quad c^T \theta \\ & \text{subject to} \quad f(\theta, q_d^{(i)}) \leq 0, \quad i = 1, \dots, N_k. \end{aligned} \tag{10}$$

   - **If** the optimization problem (10) is not feasible, the original problem (1) is not feasible as well.
   - **Else if** the last iteration is reached $(k = k_t)$, $\widehat{\theta}_{N_k}$ is a probabilistic solution to (1) with confidence δ and accuracy ε, and **Exit**.
   - **Else**, continue to the next step.

4) VALIDATION

   - Draw

   $$M_k > \left\lceil \frac{\alpha \ln k + \ln\left(\mathcal{S}_{k_t}(\alpha)\right) + \ln \frac{1}{\delta}}{\ln\left(\frac{1}{1-\varepsilon}\right)} \right\rceil \tag{11}$$

   i.i.d. samples $\mathbf{q}_v = \{q_v^{(1)} \dots q_v^{(M_k)}\}$ from the uncertainty set $\mathbb{Q}$ based on the underlying distribution. The parameter $\mathcal{S}_{k_t}(\alpha)$ in (11) is a finite hyperharmonic series $\mathcal{S}_{k_t}(\alpha) = \sum_{k=1}^{k_t} \frac{1}{k^\alpha}$.
   - **If** $\mathbb{I}_f(\widehat{\theta}_{N_k}, q_v^{(i)}) = 0$ for $i = 1, \dots, M_k$; **then**, $\widehat{\theta}_{N_k}$ is a probabilistic solution to (1) with confidence δ and accuracy ε, and **Exit**.
   - **Else**, goto step (2).

Theorem 2: Suppose that Assumptions 1 and 2 hold. If at iteration k Algorithm 2 exits with a probabilistic solution $\widehat{\theta}_{N_k}$, then it holds that $V(\widehat{\theta}_{N_k}) \leq \varepsilon$ with probability no smaller than $1-\delta$

$$\Pr\left\{V(\widehat{\theta}_{N_k}) \leq \varepsilon\right\} \geq 1-\delta.$$

Proof: See Appendix B. ■

Remark 2 (Choice of N_{k_t}): The cardinality of the design sample set at the last iteration in Algorithms 1 and 2 N_{k_t}, is chosen to be exactly equal to the bounds (6) and (9) respectively. Therefore, the complexity of the last iteration, if it is reached, is exactly equal to that of the scenario approach and the scenario with discarded constraints.

Remark 3 (Value of the Objective): If Algorithms 1 and 2 have a successful exit at iteration $k < k_t$, it implies that the number of samples used for design is smaller than the number used in the scenario approach or scenario with discarded constraints. Note that the consequent reduction in the number of design samples may potentially improve the objective value $c^T\widehat{\theta}_{N_k}$, with respect to the one obtained by the scenario approach.

Algorithm 2 is different from the algorithm presented in [2], which was derived for non-convex problems, in a number of aspects. That is, the cardinality of the sequence of sample sets used for design and validation increase linearly with iteration counter k, while they increase exponentially in [2]. Furthermore, the cardinality of the validation sample set at the last iteration M_{k_t} in [2] is chosen to be equal to the cardinality of the sample set used for design at the last iteration N_{k_t} while, in the presented algorithm M_{k_t} and hence β_w are chosen based on the additive Chernoff bound which is less conservative.

We also note that both Algorithms 1 and 2 falls within the class of SPV algorithms in which the "design" and "validation" steps are independent. As a result, in principle we could use the same strategy as Algorithm 1 to tackle discarded constraints problems. Nevertheless, Algorithm 2 appears to be more suitable for discarded constraints problems, since (13) forces the solution to violate some constraints.

C. Termination Parameter k_t

The termination parameter k_t defines the maximum number of iterations of the algorithm which can be chosen by the user. We note that the choice of k_t directly affects the cardinality of the sample sets used for design N_k and validation M_k at each iteration, although they converge

Algorithm 2 SEQUENTIAL RANDOMIZED ALGORITHM: PARTIAL CONSTRAINT SATISFACTION

1) INITIALIZATION

   Set the iteration counter to zero $(k = 0)$. Choose the desired probabilistic levels ε, δ, the desired number of iterations $k_t > 1$, the desired number of discarded constraints r and define the following parameters:

$$\beta_v \doteq \max\left\{1, \beta_w \left(k_t \ln \frac{k_t}{\delta}\right)^{-1}\right\}, \quad \beta_w \doteq \frac{1}{4\varepsilon} \ln \frac{1}{\delta}. \tag{12}$$

2) UPDATE

   Set $k = k + 1$, $N_k = \left\lceil N \frac{k}{k_t} \right\rceil$ and $N_{k,r} = \left\lceil \frac{(N-r)k}{k_t} \right\rceil$ where N is chosen based on (9).

3) DESIGN

   - Draw N_k i.i.d. samples $\mathbf{q}_d = \{q_d^{(1)} \ldots q_d^{(N_k)}\}$ from the uncertainty set $\mathbb{Q}$ based on the underlying distribution.
   - Solve the following random convex program:

$$\widehat{\theta}_{N_k} = \arg \underset{\theta \in \Theta}{\text{minimize}} \quad c^T \theta \tag{13}$$

$$\text{subject to} \quad f(\theta, q_d^{(i)}) \leq 0, \quad i = 1, \ldots, N_{k,r}.$$

   - **If** the optimization problem (13) is not feasible, the original problem (1) is not feasible as well.
   - **Else if** the last iteration is reached $(k = k_t)$, $\widehat{\theta}_{N_k}$ is a probabilistic solution to (1) with confidence δ and accuracy ε, and **Exit**.
   - **Else**, continue to the next step.

4) VALIDATION

   - Draw $M_k = \left\lceil 2k\beta_v \frac{1}{\varepsilon} \ln \frac{k_t}{\delta} \right\rceil$ i.i.d. samples $\mathbf{q}_v = \{q_v^{(1)} \ldots q_v^{(M_k)}\}$ from the uncertainty set $\mathbb{Q}$ based on the underlying distribution.
   - **If**

$$\frac{1}{M_k} \sum_{i=1}^{M_k} \mathbb{I}_f(\widehat{\theta}_{N_k}, q_v^{(i)}) \leq \left(1 - (k\beta_v)^{-1/2}\right) \varepsilon \tag{14}$$

     then, $\widehat{\theta}_{N_k}$ is a probabilistic solution to (1) with confidence δ and accuracy ε, and **Exit**.
   - **Else**, goto step (2).

to fixed values (independent of k_t) at the last iteration. In problems for which the bounds (6) and (9) are large, we would suggest to use larger k_t. Then, the sequence of sample bounds N_k starts from a smaller number and does not increase significantly with the iteration counter k. We also remark that the right hand side of the inequality (14) in Algorithm 2 cannot be negative which in turn requires β_v to be greater than one. This condition is taken into account in defining β_v in (12). However, we can avoid generating $\beta_v < 1$ by the appropriate choice of k_t. To this end, we solve the inequality $\beta_v \geq 1$ for k_t as follows:

$$\beta_v \doteq \beta_w \left(k_t \ln \frac{k_t}{\delta} \right)^{-1} \geq 1 \Longrightarrow k_t \ln \frac{k_t}{\delta} \leq \beta_w \Longrightarrow \frac{k_t}{\delta} \ln \frac{k_t}{\delta} \leq \frac{\beta_w}{\delta}.$$

For implementation purposes, it is useful to use the function “LambertW” also known as “Omega function” or “product logarithm”[2]. Then, we solve the previous inequality for k_t

$$k_t \leq \frac{\beta_w}{\text{LambertW}\left(\frac{\beta_w}{\delta}\right)}.$$

D. Complexity Analysis

The sample complexity (and computational complexity) of Algorithms 1 and 2 is a random variable because the number of iterations is random. The sample complexity in which the algorithm terminates (N_k and M_k) is only known *a posteriori*, while in the scenario approach we can establish *a priori* sample bounds. Taking into account that the asymptotic computational complexity of most SDP solvers increase exponentially with the size of the problem under consideration [16], we conclude that if Algorithms 1 and 2 exit with smaller number of design samples than the bounds (6) and (9), which is the case most of the times, the reduction in the number of design samples can significantly improve the computational complexity. For instance, asymptotic computational complexity of solving a $n \times n$ linear matrix inequality (LMI) with m decision variables using the SDP solver SEDUMI is of order $O(m^2 n^{2.5} + n^{3.5})$ [16]. Hence, decrease in n, which represents the size of LMI, can significantly reduce the computational complexity. We also note that the computational complexity of validation steps in both presented algorithms is not significant since they just require *analysis* of a candidate solution for a number of i.i.d. samples extracted from the uncertainty set.

[2] This function is the inverse function of $f(W) = We^W$. In other words, $W = \text{LambertW} f(W)$; see e.g. [14] for more details. The Matlab command is `W = lambertw(f(W))`.

ε	δ	k_t	*Scenario Bound*	*Design Samples*			*Validation Samples*			*Objective Value*			*Iteration Number*		
				Mean	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*
0.2	10^{-2}	20	1238	208.94	92.69	496	34	0.17	35	0.6104	0.006	0.6224	3.37	1.49	8
0.1	10^{-4}	20	2548	555.9	237.7	1274	115.6	0.54	117	0.6175	0.004	0.6274	4.36	1.86	10
0.05	10^{-6}	30	5240	954	346.2	2096	334.66	0.68	336	0.6209	0.03	0.6279	5.46	1.96	12

TABLE I

SIMULATION RESULTS OBTAINED USING ALGORITHM 1

ε	δ	k_t	*Scenario Bound*	*Design Samples*			*Validation Samples*			*Objective Value*			*Iteration Number*		
				Mean	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*	*Mean*	*Standard Deviation*	*Worst Case*
0.2	10^{-2}	20	1238	209	33.8	310	257.8	41.5	381	0.6121	0.004	0.6229	3.38	0.54	5
0.1	10^{-4}	20	2548	436.3	90.5	513	733	172.5	977	0.6278	0.01	0.6434	3	0.707	4
0.05	10^{-6}	30	5240	567.6	115.3	774	1515.8	308.1	2067	0.6303	0.02	0.742	2.13	1.15	6

TABLE II

SIMULATION RESULTS OBTAINED USING ALGORITHM 2

IV. APPLICATION TO HARD DISK DRIVE SERVO DESIGN

In this section, we employ the developed algorithms to solve a non-trivial industrial example. The problem under consideration is the design of a robust track following controller for a hard disk drive (HDD) servo system affected by parametric uncertainty. Servo system in HDD plays a crucial role in increasing the storage capacity by providing a more accurate positioning algorithm. The goal in this application is to achieve the storage density of 10 Tera bit per square inch $(10Tb/in^2)$. It requires the variance of the deviation of read/write head from the center of a data track to be less than 1.16 nanometer. Such a high performance has to be achieved in a robust manner, that is, for all drives produced in a mass production line. On the other hand, some imperfections in the production line such as manufacturing tolerances and slightly different materials or environmental conditions lead to slightly different dynamics over a batch of products.

A voice coil motor (VCM) actuator in a disk drive system can be modeled in the form

$$P_{VCM} = \sum_{i=1}^{3} \frac{A_i}{s^2 + 2\zeta_i \omega_i s + \omega_i^2} \tag{15}$$

where ζ_i, ω_i and A_i are damping ratio, natural frequency and modal constant for each resonance mode, see [13] for their nominal values. We assume each natural frequency, damping ratio and modal constant to vary by 5%, 5% and 10% from their nominal values respectively. Hence, there are nine uncertain parameters in the plant. Since the problem is of regulation type, the sensitivity transfer function is of vital importance. In order to shape the sensitivity transfer function, we augment the open-loop plant with necessary weighting functions. The objective is to design a full order dynamic output feedback controller which minimizes the worst case $\mathcal{H}_\infty$ norm of the transfer function from disturbance to output. Neglecting the uncertain parameters, this problem can be easily reformulated in terms of linear matrix inequalities [18]. Uncertain parameters enter into the plant description in a non-affine fashion; therefore, classical robust techniques are unable to solve the problem without introducing conservatism.

The sequential algorithms of Section III are implemented in Matlab using the toolbox Randomized Algorithm Control Toolbox (RACT) [20]. In the simulations, we assumed the probability density function of all uncertain parameters to be uniform. The choice of uniform distribution is chosen due to its worst case nature [4]. The number of discarded constraints r in Algorithm 2 is chosen to be zero. The resulting optimization problem is solved for different values of ε, δ and k_t. Furthermore, we run the simulation 100 times for each pair. The mean, standard deviation and worst case values of the number of design samples, validation samples, objective value and the iteration number in which the algorithm exits are tabulated in Table I and Table II. The scenario bounds are also shown in the same tables for an easy comparison. It is observed that using the proposed algorithms, we can achieve the same probabilistic levels with much smaller number of design samples.

V. Conclusions

We proposed two new sequential methods for solving in a computational efficient way uncertain convex optimization problems. The main philosophy behind the proposed sequential randomized algorithms stems from the consideration that it is easy, from a computational viewpoint, to validate a given “candidate solution” for a large number of random samples. The algorithms have been tested on a numerical example, and extensive numerical simulations show how the total computational effort is "diluted" by applying the proposed sequential methodology.

ACKNOWLEDGEMENTS

The results presented in this paper were obtained when the first author was visiting CNR-IEIIT for a period of six months under DSI funding.

APPENDIX A
PROOF OF THE THEOREM 1

Following the same reasoning as in [17], we introduce the following events

$$\text{Iter}_k \doteq \{\text{the } k\text{th outer iteration is reached}\},$$

$$\text{Feas}_k \doteq \{\widehat{\theta}_{N_k} \text{ is declared as feasible in the "validation" step}\},$$

$$\text{Bad}_k \doteq \{V(\widehat{\theta}_{N_k}) > \varepsilon\},$$

$$\text{ExitBad}_k \doteq \{\text{Algorithm 1 exits at iteration } k \cap \text{Bad}_k\},$$

$$\text{ExitBad} \doteq \{\text{Algorithm 1 exits at some unspecified iteration } k \cap \text{Bad}_k\}.$$

The goal is to bound the probability of the event "ExitBad". Since $\text{ExitBad}_i \cap \text{ExitBad}_j = \emptyset$ for $i \neq j$, the probability of the event "ExitBad" can be reformulated in terms of the event "ExitBad_k" as

$$\begin{aligned}\Pr\{\text{ExitBad}\} &= \Pr\{\text{ExitBad}_1 \cup \text{ExitBad}_2 \cup \cdots \cup \text{ExitBad}_{k_t}\}\\ &= \Pr\{\text{ExitBad}_1\} + \Pr\{\text{ExitBad}_2\} + \cdots + \Pr\{\text{ExitBad}_{k_t}\}. \qquad (16)\end{aligned}$$

From the definition of the event "ExitBad_k" and by considering the point that to exit at iteration k, Algorithm 1 needs to reach kth iteration and declares $\widehat{\theta}_{N_k}$ as feasible, we arrive at

$$\begin{aligned}\Pr\{\text{ExitBad}_k\} &= \Pr\{\text{Feas}_k \cap \text{Bad}_k \cap \text{Iter}_k\} = \Pr\{\text{Feas}_k \cap \text{Bad}_k \mid \text{Iter}_k\}\Pr\{\text{Iter}_k\}\\ &\leq \Pr\{\text{Feas}_k \cap \text{Bad}_k \mid \text{Iter}_k\} = \Pr\{\text{Feas}_k \mid \text{Bad}_k \cap \text{Iter}_k\}\Pr\{\text{Bad}_k \mid \text{Iter}_k\}\\ &\leq \Pr\{\text{Feas}_k \mid \text{Bad}_k \cap \text{Iter}_k\}. \qquad (17)\end{aligned}$$

Using the result of Theorem 1 in [8], we can bound the right hand side of (17)

$$\Pr\{\text{Feas}_k \mid \text{Bad}_k \cap \text{Iter}_k\} < (1-\varepsilon)^{M_k}. \qquad (18)$$

Combining (16) and (18) results in

$$\Pr\{\text{ExitBad}\} < (1-\varepsilon)^{M_1} + (1-\varepsilon)^{M_2} + \cdots + (1-\varepsilon)^{M_{k_t}} = \sum_{k=1}^{k_t} (1-\varepsilon)^{M_k}. \tag{19}$$

The summation in (19) can be made arbitrary small by the appropriate choice of M_k. By choosing

$$(1-\varepsilon)^{M_k} = \frac{1}{k^\alpha}\frac{1}{\mathcal{S}_{k_t}(\alpha)}\delta \tag{20}$$

where $\delta \in (0,1)$ is a (small) desired probability level, we have

$$\sum_{k=1}^{k_t}(1-\varepsilon)^{M_k} = \sum_{k=1}^{k_t}\frac{1}{k^\alpha}\frac{1}{\mathcal{S}_{k_t}(\alpha)}\delta = \frac{1}{\mathcal{S}_{k_t}(\alpha)}\delta\sum_{k=1}^{k_t}\frac{1}{k^\alpha} = \frac{1}{\mathcal{S}_{k_t}(\alpha)}\delta\mathcal{S}_{k_t}(\alpha) = \delta.$$

Therefore, the appropriate choice of M_k which guarantees $\Pr\{\text{ExitBad}\} < \delta$ can be computed by solving (20) for M_k which results in the bound (11).

Appendix B
Proof of the Theorem 2

Given N, $\varepsilon \in (0,1)$, $\rho \in [0,1)$ and $f : \Theta \times \mathbb{Q} \to \mathbb{R}$, the probability of one-sided constrained failure, denoted by $p_f(N,\varepsilon,\rho)$ is defined as

$$p_f(N,\varepsilon,\rho) \doteq \Pr\left\{\mathbf{q} \in \mathbb{Q}^N : \text{there exist } \theta \in \Theta \text{ such that } \widehat{V}(\theta,\mathbf{q}) \le \rho \text{ and } V(\theta) - \widehat{V}(\theta,\mathbf{q}) > \varepsilon\right\}.$$

Denote by δ_k the probability of miss-classification at iteration k. Therefore,

$$\delta_k \le \Pr\left\{\mathbf{q}_v \in \mathbb{Q}^{M_k} : \widehat{V}(\widehat{\theta}_{N_k},\mathbf{q}_v) \le \left(1-(k\beta_v)^{-1/2}\right)\varepsilon \text{ and } V(\widehat{\theta}_{N_k}) > \varepsilon\right\}.$$

By defining $\rho_k \doteq \left(1-(k\beta_v)^{-1/2}\right)\varepsilon$ and $\varepsilon_k \doteq (k\beta_v)^{-1/2}\varepsilon$, the probability of miss-classification can be expressed in terms of the probability of one-sided constrained failure

$$\delta_k \le \Pr\left\{\mathbf{q}_v \in \mathbb{Q}^{M_k} : \widehat{V}(\widehat{\theta}_{N_k},\mathbf{q}_v) \le \rho_k \text{ and } V(\widehat{\theta}_{N_k}) - \widehat{V}(\widehat{\theta}_{N_k},\mathbf{q}_v) > \varepsilon_k\right\}.$$

Using Theorem 1 in [2], it follows that

$$\delta_k \le \Pr\left\{\mathbf{q} \in \mathbb{Q}^{M_k} : \frac{V(\widehat{\theta}_{N_k}) - \widehat{V}(\widehat{\theta}_{N_k},\mathbf{q}_v)}{\sqrt{V(\widehat{\theta}_{N_k})}} > \frac{\varepsilon_k}{\sqrt{\varepsilon_k + \rho_k}}\right\}. \tag{21}$$

For fixed θ, ε and any $f : \Theta \times \mathbb{Q} \to \mathbb{R}$, one-sided multiplicative Chernoff inequality is defined as

$$\Pr\{V(\theta) - \widehat{V}(\theta,\mathbf{q}) \ge \varepsilon V(\theta)\} \le e^{\frac{-V(\theta)N\varepsilon^2}{2}}. \tag{22}$$

Letting ε to be $\frac{\frac{\varepsilon_k}{\sqrt{\varepsilon_k+\rho_k}}}{\sqrt{V(\widehat{\theta}_{N_k})}}$, combining inequalities (21) and (22), and taking natural logarithm of both sides, we obtain

$$\ln \delta_k \leq \frac{-\varepsilon_k M_k}{2(\varepsilon_k + \rho_k)} = \frac{-(k\beta_v)^{-1}\varepsilon^2}{2\varepsilon} 2k\beta_v \frac{1}{\varepsilon} \ln \frac{k_t}{\delta} = \ln \frac{\delta}{k_t}.$$

Therefore, taking into account that the probability of miss-classification of the algorithm is the summation of the probability of miss-classifications at each iterations (δ_k), we arrive at

$$\sum_{k=1}^{k_t} \delta_k \leq \sum_{k=1}^{k_t} \frac{\delta}{k_t} = \frac{\delta k_t}{k_t} = \delta$$

which proves the statement.

References

[1] T. Alamo, A. Luque, D.R. Ramirez, and R. Tempo. Randomized control design through probabilistic validation. In *Proc. American Control Conference (ACC)*, pages 839–844, 2012.

[2] T. Alamo, R. Tempo, and E.F. Camacho. Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems. *IEEE Transactions on Automatic Control*, 54:2545–2559, 2009.

[3] T. Alamo, R. Tempo, and A. Luque. On the sample complexity of probabilistic analysis and design methods. In J. Willems, S. Hara, Y. Ohta, and H. Fujioka, editors, *Perspectives in Mathematical System Theory, Control, and Signal Processing*, volume 398 of *Lecture Notes in Control and Information Sciences*, pages 39–50. Springer Berlin / Heidelberg, 2010.

[4] E. Bai, R. Tempo, and M. Fu. Worst-case properties of the uniform distribution and randomized algorithms for robustness analysis. *Mathematics of Control, Signals, and Systems*, 11:183–196, 1998.

[5] G.C. Calafiore. Random convex programs. *SIAM Journal on Optimization*, 20:3427–3464, 2010.

[6] G.C. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102:25–46, 2004.

[7] G.C. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51:742–753, 2006.

[8] G.C. Calafiore and F. Dabbene. A probabilistic analytic center cutting plane method for feasibility of uncertain LMIs. *Automatica*, 43:2022–2033, 2007.

[9] G.C Calafiore, F. Dabbene, and R. Tempo. Research on probabilistic methods for control system design. *Automatica*, 47:1279–1293, 2011.

[10] M.C. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM J. on Optimization*, 19:1211–1230, 2008.

[11] M.C. Campi and S. Garatti. A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. 148(2):257–280, 2011 (preliminary version available on Optimization Online, 2008).

[12] M. Chamanbaz, F. Dabbene, R. Tempo, V. Venkataramanan, and Q-G. Wang. Sequential randomized algorithms for sampled convex optimization. In *Proc. IEEE Multi-Conference on Systems and Control (MSC)*, Hyderabad, India, 2013. Submitted.

[13] M. Chamanbaz, E. Keikha, V. Venkataramanan, Q-G Wang, and A. Al Mamun. Probabilistic robust approach for discrete multi-objective control of track-following servo systems in hard disk drives. In *Proc. 7th IFAC Symposium on Robust Control Design*, Aalborg, Denmark, 2012.

[14] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the LambertW function. *Advances in Computational Mathematics*, 5:329–359, 1996.

[15] F. Dabbene, P. S. Shcherbakov, and B. T. Polyak. A randomized cutting plane method with probabilistic geometric convergence. *SIAM Journal on Optimization*, 20:3185–3207, 2010.

[16] Y. Labit, D. Peaucelle, and D. Henrion. SEDUMI interface 1.02: a tool for solving LMI problems with SEDUMI. In *Proc. IEEE International Symposium on Computer Aided Control System Design*, pages 272–277, 2002.

[17] Y. Oishi. Polynomial-time algorithms for probabilistic solutions of parameter-dependent linear matrix inequalities. *Automatica*, 43:538–545, 2007.

[18] C. Scherer, P. Gahinet, and M. Chilali. Multiobjective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, 42:896–911, 1997.

[19] R. Tempo, G.C. Calafiore, and F. Dabbene. *Randomized Algorithms for Analysis and Control of Uncertain Systems: With Applications*. Springer, 2nd edition, 2013.

[20] A. Tremba, G.C. Calafiore, F. Dabbene, E. Gryazina, B. Polyak, P. Shcherbakov, and R. Tempo. RACT: randomized algorithms control toolbox for MATLAB. In *Proc. 17th World Congress of IFAC, Seoul*, pages 390–395, 2008.

[21] T. Wada and Y. Fujisaki. Sequential randomized algorithms for robust optimization. In *Proc. IEEE Conference on Decision and Control*, pages 6190–6195, 2007.